

JC854 U.S. PTO
06/21/00

06-23-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Xu et al.
Applicant..... Microsoft Corporation
Attorney's Docket No. MS1-554US
Title: Video Coding System and Method Using 3-D Discrete Wavelet Transforms and Entropy Coding
With Motion Information

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Box Patent Application
Commissioner of Patents and Trademarks,
Washington, D.C. 20231

From: Lewis C. Lee (Tel. 509-324-9256; Fax 509-323-8979)
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Specification—title page, plus 53 pages, including claims 1-82 and Abstract
2. Transmittal letter including Certificate of Express Mailing
3. 8 Sheets Formal Drawings (Figs. 1-13)
4. Return Post Card

Large Entity Status ☒ [x]

Small Entity Status ☐ []

Date: June 21, 2000

By: [Signature]

Lewis C. Lee
Reg. No. 34,656

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL6 2435 2065

Date: June 21, 2000

By: [Signature]

Helen M. Hare

JC829 U.S. PTO
09/599160
06/21/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Video Coding System and Method Using 3-D Discrete
Wavelet Transform and Entropy Coding With Motion
Information**

Inventor(s):
Jizheng Xu
Shipeng Li
Ya-Qin Zhang

ATTORNEY'S DOCKET NO. MS1-554US

1 **TECHNICAL FIELD**

2 This invention relates to systems and methods for video coding. More
3 particularly, this invention relates to systems and methods that employ wavelet
4 transforms for video coding.
5

6 **BACKGROUND**

7 Efficient and reliable delivery of video data is becoming increasingly
8 important as the Internet continues to grow in popularity. Video is very appealing
9 because it offers a much richer user experience than static images and text. It is
10 more interesting, for example, to watch a video clip of a winning touchdown or a
11 Presidential speech than it is to read about the event in stark print.

12 With the explosive growth of the Internet and fast advance in hardware
13 technologies and software developments, many new multimedia applications are
14 emerging rapidly. Although the storage capability of the digital devices and the
15 bandwidth of the networks are increasing rapidly, video compression still plays an
16 essential role in these applications due to the exponential growth of the multimedia
17 contents both for leisure and at work. Compressing video data prior to delivery
18 reduces the amount of data actually being transferred over the network. Image
19 quality is lost as a result of the compression, but such loss is generally tolerated as
20 necessary to achieve acceptable transfer speeds. In some cases, the loss of quality
21 may not even be detectable to the viewer.

22 Many emerging applications require not only high compression efficiency
23 from the various coding techniques, but also greater functionality and flexibility.
24 For example, in order to facilitate content-based media processing, retrieval and
25 indexing, and to support user interaction, object-based video coding is desired. To

enable video delivery over heterogeneous networks (e.g., the Internet) and wireless channels, error resilience and bit-rate scalability are required. To produce a coded video bitstream that can be used by all types of digital devices, regardless of their computational, display and memory capabilities, both resolution scalability and temporal scalability are needed.

One common type of video compression is the motion-compensation-based video coding scheme, which is employed in essentially all compression standards such as MPEG-1, MPEG-2, MPEG-4, H.261, and H.263. Such video compression schemes use predictive approaches that encode information to enable motion prediction from one video frame to the next.

Unfortunately, these conventional motion-compensation-based coding systems, primarily targeted for high compression, fail to provide new functionalities such as scalability and error robustness. The recent MPEG-4 standard adopts an object-based video coding scheme to enable user interaction and content manipulation, but the scalability of MPEG-4 is very limited. Previously reported experiments with MPEG-2, MPEG-4, and H.263 indicate that the coding efficiency generally loses 0.5-1.5dB with every layer, compared with a monolithic (non-layered) coding scheme. See, for example, B. G. Haskell, A. Puri and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, New York, 1997; and L. Yang, F. C. M. Martins, and T. R. Gardos, "Improving H.263+ Scalability Performance for Very Low Bit Rate Applications," In *Proc. Visual Communications and Image Processing*, San Jose, CA, January 1999, SPIE.

Since these standard coders are all based on a predictive structure, it is difficult for the coding schemes to achieve efficient scalability due to the drift

1 problem associated with predictive coding. Currently, there are proposals for
2 MPEG-4 streaming video profile on fine granularity scalable video coding.
3 However, these proposals are limited to provide flexible rate scalability only and
4 the coding efficiency is still much lower than that of non-layered coding schemes.

5 An alternative to predictive-based video coding schemes is three
6 dimensional (3-D) wavelet video coding. One advantage of 3-D wavelet coding
7 over predictive video coding schemes is the scalability (including rate, PSNR,
8 spatial, and temporal), which facilitates video delivery over heterogeneous
9 networks (e.g., the Internet) and future wireless video services. However,
10 conventional 3-D wavelet coding does not use motion information that is proven
11 to be very effective in predictive coders in terms of removing temporal
12 redundancy. Although the computationally intensive motion estimation is
13 avoided, the performance of 3D wavelet video coding remains very sensitive to
14 the motion. Without motion information, motion blur occurs due to a temporal
15 averaging effect of several frames. In addition, most 3-D wavelet video coders do
16 not support object-based functionality, which is needed in the next generation
17 multimedia applications.

18 Accordingly, there is a need for an efficient 3-D wavelet transform for
19 video coding that employs motion information to reduce the sensitivity to motion
20 and remove the motion blur in the resulting video playback. Additionally, an
21 improved 3-D wavelet transform should support object-based functionality.

22 23 SUMMARY

24 A video encoding system and method utilizes a three-dimensional (3-D)
25 wavelet transform and entropy coding that utilize motion information in a way to

1 reduce the sensitivity to motion and remove any motion blur in the resulting video
2 playback.

3 In one implementation, the video encoding process initially estimates
4 motion trajectories of pixels in a video object from frame to frame in a video
5 sequence. The motion estimation accounts for motion of the video object
6 throughout the frames, effectively aligning the pixels in the time direction. The
7 motion estimation may be accomplished by matching corresponding pixels in the
8 video object from frame to frame.

9 After motion estimation, a 3-D wavelet transform is applied in two parts.
10 First, a temporal 1-D wavelet transform is applied to the corresponding pixels
11 along the motion trajectories in a time direction. The temporal wavelet transform
12 produces decomposed frames of temporal wavelet transforms, where the spatial
13 correlation within each frame is well preserved. Second, a spatial 2-D wavelet
14 transform is applied to all frames containing the temporal wavelet coefficients.
15 The wavelet transforms produce coefficients within different sub-bands.

16 The process then codes wavelet coefficients. In particular, the coefficients
17 are assigned various contexts based on the significance of neighboring samples in
18 previous, current, and next frame, thereby taking advantage of any motion
19 information between frames. The wavelet coefficients are coded independently
20 for each sub-band to permit easy separation at a decoder, making resolution
21 scalability and temporal scalability natural and easy. During the coding, bits are
22 allocated among sub-bands according to a technique that optimizes rate-distortion
23 characteristics. In one implementation, the number of bits are truncated at points
24 in a rate-distortion curve that approximates a convex hull of the curve.
25

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a video distribution system, including a video encoder at a content producer/provider and a video decoder at a client.

Fig. 2 is a flow diagram of a video coding process using three-dimensional shape-adaptive discrete wavelet transforms and motion estimation information.

Fig. 3 illustrates four frames in a video sequence to demonstrate motions estimation of pixels from frame to frame.

Fig. 4 illustrates two consecutive frames and demonstrates a case where a pixel continues from one frame to the next.

Fig. 5 illustrates two consecutive frames and demonstrates a case where a pixel terminates in a current frame and does not continue to the next frame.

Fig. 6 illustrates two consecutive frames and demonstrates a case where a pixel emerges in the next frame, but does not appear in the current frame.

Fig. 7 illustrates two consecutive frames and demonstrates a case where to pixels in the current frame collide at one pixel in the next frame.

Fig. 8 is a flow diagram of a 3-D wavelet transform process applied to video frames.

Fig. 9 illustrates sub-bands within a video frame that are formed by the wavelet transform.

Fig. 10 is a flow diagram of a sub-band encoding process.

Fig. 11 illustrates three frames to demonstrate how a context for a pixel is determined in terms of neighboring pixels.

Fig. 12 is a flow diagram of a bitstream construction and truncation process.

Fig. 13 illustrates a rate-distortion curve that is used in the Fig. 12 process.

1
2 **DETAILED DESCRIPTION**

3 This disclosure describes a video coding scheme that utilizes a three-
4 dimensional (3-D) wavelet transform and coding scheme that is suitable for
5 object-based video coding. The 3-D wavelet transform uses motion trajectories in
6 the temporal direction to obtain more efficient wavelet decomposition and to
7 reduce or remove the motion blurring artifacts for low bit-rate coding.

8 The 3-D wavelet transformation produces coefficients within different sub-
9 bands. An entropy coder is employed to code each sub-band independently in a
10 manner that takes advantage of the motion information. The entropy coder also
11 uses rate-distortion curves to optimize the bit-allocation among sub-bands. Given
12 these attributes, the entropy coder process may be referred to as "Embedded Sub-
13 band Coding with Optimized Truncation" (or short handedly as "ESCOT"). The
14 entropy coder outputs independent embedded bitstreams for each sub-band that
15 meet scalability requirements of new multimedia applications.

16 Accordingly, unlike conventional 3-D wavelet coding schemes, motion
17 information is used for both 3-D shape adaptive wavelet transforms and the
18 entropy coding. The proposed coding scheme has comparable coding efficiency
19 with MPEG4, while having more functionalities and flexibility, such as, flexible
20 rate scalability, spatial scalability, and temporal scalability. This makes the coding
21 scheme very suitable for numerous applications like video streaming, interactive
22 multimedia applications, and video transmission over wireless channels.

23 The coding scheme is described in the context of delivering video data over
24 a network, such as the Internet or a wireless network. However, the video coding
25 scheme has general applicability to a wide variety of environments.

001250 09T5550

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Exemplary System Architecture

Fig. 1 shows a video distribution system 100 in which a content producer/provider 102 produces and/or distributes video over a network 104 to a client 106. The network 104 is representative of many different types of networks, including cable, the Internet, a LAN (local area network), a WAN (wide area network), a SAN (storage area network), and wireless networks (e.g., satellite, cellular, RF, microwave, etc.).

The content producer/provider 102 may be implemented in many ways, including as one or more server computers configured to store, process, and distribute video data. The content producer/provider 102 has a video storage 110 to store digital video files 112 and a distribution server 114 to encode the video data and distribute it over the network 104. The server 104 has one or more processors 120, an operating system 122 (e.g., Windows NT, Unix, etc.), and a video encoder 124. The video encoder 124 may be implemented in software, firmware, and/or hardware. The encoder is shown as a separate standalone module for discussion purposes, but may be constructed as part of the processor 120 or incorporated into operating system 122 or other applications (not shown).

The video encoder 124 encodes the video data stored as files 112 using a 3-D wavelet transformer 130 and codes the resulting coefficients using an entropy coder 132. In one implementation, the 3-D wavelet transformer 130 implements a shape-adaptive discrete wavelet transform (SA-DWT), which is an efficient wavelet transform for arbitrarily shaped visual objects. With SA-DWT, the number of coefficients after SA-DWT is identical to the number of pixels in an original arbitrarily shaped visual object. In addition, SA-DWT preserves the

1 spatial correlation, locality properties of wavelet transforms, and self-similarity
2 across sub-bands. It is noted, however, that aspects of this invention may be
3 implemented using other types of wavelet transforms.

4 The video encoder 124 utilizes motion information in the temporal
5 direction of the video sequence. A motion trajectory for each pixel inside a video
6 object is traced from frame-to-frame using one of a variety of motion estimation
7 processes. Then, a one-dimensional (1-D) SA-DWT is performed along each
8 motion trajectory in the time direction to produce temporally decomposed frames
9 of wavelet coefficients. After temporal decomposition, a spatial two-dimensional
10 (2-D) SA-DWT is applied to all temporally decomposed frames.

11 The 3-D (i.e., 1-D temporal and 2-D spatial) wavelet transform solves two
12 problems. First, it can handle arbitrarily shaped video objects while having
13 flexible bit-rate, spatial, and temporal scalabilities as in most wavelet-based
14 coding schemes. Secondly, the 3-D wavelet transform tracks the video object
15 motion and performs the wavelet transform among corresponding pixels for that
16 object while keeping the spatial correlation within a frame. Thus, it will
17 efficiently decompose the video-object sequence and more efficient compression
18 is feasible.

19 After the wavelet transform, the entropy coder 132 codes the coefficients of
20 each sub-band independently. The coder assigns various contexts to the
21 coefficients based on data regarding neighboring samples in the previous, current,
22 and next frames. This context assignment thus takes advantage of the motion
23 information between frames. The coded bitstreams for each sub-band are
24 subsequently combined to form a final bitstream that satisfies scalability
25

1 requirements. In one implementation, the coded bitstreams are combined using a
2 multi-layer bitstream construction technique.

3 The client 106 may be embodied in many different ways, including as a
4 computer, a handheld device, a set-top box, a television, a game console,
5 information appliance, wireless communication device, and so forth. The client
6 106 is equipped with a processor 140, a memory 142, and one or more media
7 output devices 144. The memory 142 stores an operating system 150 (e.g., a
8 Windows-brand operating system) that executes on the processor 140.

9 The operating system 150 implements a client-side video decoder 152 to
10 decode the video stream. The decoder employs an inverse wavelet transformer
11 154 to decode the video stream. Following decoding, the client stores the video in
12 memory 142 and/or plays the video via the media output devices 144.

13 Coding Process

14 Fig. 2 shows a video coding process 200 for coding video objects. The
15 process 200 may be implemented, for example, by the video encoder 124 in the
16 content producer/provider 102. The process may be implemented as computer-
17 readable instructions stored in a computer-readable medium (e.g., memory,
18 transmission medium, etc.) that, when executed, perform the operations illustrated
19 as blocks in Fig. 2.

20 At block 202, the video encoder estimates motion trajectories of pixels in a
21 video object from frame to frame in a video sequence to account for motion of the
22 video object throughout the frames. In one implementation, the video encoder
23 uses a pixel matching process to match corresponding pixels from frame-to-frame
24 in the temporal direction. The matching operation traces motion trajectories for
25

1 the corresponding pixels, thereby aligning the pixels in the temporal direction. It
2 is noted that other motion estimation schemes may be used instead of the pixel
3 matching process.

4 At block 204, the video encoder uses the wavelet transformer 130 to
5 perform a wavelet transform on the corresponding pixels in the time dimension
6 along the motion trajectories. In one implementation, the transformer uses a
7 temporal 1-D shape-adaptive discrete wavelet transform (SA-DWT) for the
8 corresponding pixels. The temporal wavelet transform produces decomposed
9 frames of temporal wavelet transforms, where the spatial correlation within each
10 frame is well preserved.

11 At block 206, the wavelet transformer 130 applies a spatial 2-D shape-
12 adaptive discrete wavelet transform for all frames containing the temporal wavelet
13 coefficients (block 206). The wavelet transforms produce coefficients within
14 different sub-bands. The 3-D SA-DWTs of blocks 204 and 206 are explored in
15 more detail below under the heading "3-D SA-DWT (Blocks 204 and 206)".

16 At block 208, the entropy coder 132 codes the wavelet coefficients
17 independently for each sub-band and optimizes the bits allocated to each sub-band.
18 The entropy coder 132 outputs a bitstream of independently coded sub-bands. The
19 entropy encoding operation is described below in more detail under the heading
20 "ESCOT (Block 208)".

21 22 **3-D SA-DWT (Blocks 204 and 206)**

23 As shown in operation 202 of Fig. 2, the video encoder initially constructs a
24 1-D array of corresponding pixels obtained from motion estimation (e.g., pixel-
25

1 matching scheme) to identify corresponding pixels from frame to frame. The
2 motion estimation aligns the pixels the temporal direction.

3 Fig. 3 shows four frames 300, 302, 304, and 306 plotted along the time
4 dimension. Each frame has a video object 310 in the form of a “smiley face” that
5 moves from frame to frame. Consider a pixel “p” used to form an eye in the
6 smiley face object 310. The first task prior to transformation is to match the pixel
7 p in each frame to account for motion of the object. The corresponding pixels
8 from frame-to-frame are linked by line 312.

9 After the 1-D array of corresponding pixels is built, the wavelet transformer
10 130 at content provider 102 performs a temporal decomposition along the motion
11 trajectories. More specifically, the transformer 130 applies a 1-D shape-adaptive
12 discrete wavelet transform to the 1-D array to obtain a 1-D coefficient array. The
13 coefficients in the 1-D array are then redistributed to their corresponding spatial
14 position in each frame.

15 A video object normally is not limited to 2-D translation movement within
16 a frame, but may move in/out or zoom in/out of a video scene any time. This
17 gives rise to four separate cases of pixel transitions from frame to frame.

18
19 **Case 1: Continuing pixels.** This is the normal case as pixels continue in
20 one-to-one correspondence between two consecutive frames. In this case,
21 the temporal 1-D pixel array is extended to include the corresponding pixel
22 from the next frame. Fig. 4 illustrates the continuing pixel case, where a
23 pixel p continues from one frame n to a next frame n+1.
24
25

1 computer-executable instructions embodied on computer readable media (e.g.,
2 storage media, communications media, etc.).

3 Given a group of pictures/frames F_i , for $i=0, \dots, N-1$, it is assumed that the
4 motion of each pixel with reference to the next frame has been obtained using a
5 motion estimation, for example, a block-based motion estimation algorithm. For
6 each block in frame i that contains pixels from the video object, a search for the
7 best-matched block in frame $i+1$ is made and the motion vector for that block is
8 estimated. For purposes of 3-D SA-DWT, the motion vector of every pixel within
9 a block is set to the same as that of the block. Other motion estimation techniques
10 may be used.

11 After motion estimation, each pixel in the current frame may represent one
12 of the four cases described above: continuing pixels, terminating pixels, emerging
13 pixels, and colliding pixels. Additionally, all pixels in the last frame F_{N-1} are
14 terminating pixels since there is no "next" frame. For discussion purposes, assume
15 that the wavelet transformer 130 employs odd-symmetric bi-orthogonal wavelet
16 filters, although other types of wavelet filters can also be used.

17 At block 802, the transformer 130 initializes the 3-D shape-adaptive
18 discrete wavelet transform. In our example, counter value "i" is set to 0 and all
19 pixels within an object boundary in all N frames are marked as UNSCANNED.

20 At block 804, the wavelet transformer 130 performs 1-D temporal SA-
21 DWT on the frames. This operation includes constructing temporal 1-D pixel
22 arrays, transforming those arrays to produce low-pass (LP) and high-pass (HP)
23 coefficients, and organizing LP and HP coefficients into low-pass and high-pass
24 frames.
25

One preferred implementation of the 1-D temporal SA-DWT is illustrated as blocks 804(1)-804(12). The transform operation (block 804) loops through every pixel in every frame of the video sequence. Block 804(1) represents this iterative process as being for every pixel $p_i(x_i, y_i)$ within object boundary in frame F_i . At block 804(2), the pixel is examined to see if it is marked as UNSCANNED. If not, the pixel has already been considered and the process proceeds to the next pixel at block 804(1). Otherwise, assuming the pixel is still marked UNSCANNED (i.e., the “yes” branch from block 804(2)), the pixel becomes the first pixel of a new temporal 1-D pixel array (block 804(3)). Essentially, this pixel represents the emerging pixel case where it is the first pixel to originate in a frame.

The inner loop of operations consisting of blocks 804(4)-804(9) evaluate whether the pixels are continuing pixels, thereby growing the pixel array, or terminating pixels that end the array. At block 804(4), the pixel is evaluated to determine whether it is a terminating pixel, meaning that there is no corresponding pixel in the next frame. Introducing “j” as a new counter equal to “i”, if pixel $p_j(x_j, y_j)$ is a terminating pixel, it is the last pixel in the temporal 1-D array and hence the array is ready for transformation at block 804(9) (described below).

Conversely, if pixel $p_j(x_j, y_j)$ is not a terminating pixel (i.e., the “no” branch from block 804(4)), the process evaluates whether the corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ in frame F_{j+1} is marked as UNSCANNED, where $(x_{j+1}, y_{j+1}) = (x + mv_x, y + mv_y)$ and (mv_x, mv_y) is the motion vector from pixel $p_j(x_j, y_j)$ in frame F_j to its corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ in frame F_{j+1} (block 804(5)). If the corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ is UNSCANNED (i.e., the “yes” branch from block 804(5)), the corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ is added as the next pixel in the 1-D pixel array (block 804(6)). This situation represents the continuing pixel

case (Fig. 4) in which consecutive pixels are added to the temporal 1-D array. The corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ is then marked as SCANNED to signify that it has been considered (block 804(7)). Process continues with consideration of the next corresponding pixel $p_{j+2}(x_{j+2}, y_{j+2})$ in the next frame F_{j+2} (block 802(8)).

On the other hand, as indicated by the “no” branch from block 804(5), the corresponding pixel $p_{j+1}(x_{j+1}, y_{j+1})$ may have already been marked as SCANNED, indicating that this pixel also corresponded to at least one other pixel that has already been evaluated. This represents the colliding pixel case illustrated in Fig. 7. In this case, the subject pixel $p_j(x_j, y_j)$ in frame F_j will terminate the 1-D pixel array (block 804(9)).

At block 802(10), the transformer applies 1-D arbitrary length wavelet filtering to each terminated 1-D pixel array. This operation yields a transformed low-pass thread of coefficients $L_k(x_k, y_k)$, $k=i, \dots, j-1$, and a transformed high-pass thread of coefficients $H_k(x_k, y_k)$, $k=i, \dots, j-1$. The low-pass coefficients $L_k(x_k, y_k)$ are organized into a low-pass frame k at position (x_k, y_k) and the high-pass coefficients $H_k(x_k, y_k)$ are organized into a high-pass frame k at position (x_k, y_k) . Isolated pixels can be scaled by a factor (e.g., square root of 2) and put back into their corresponding positions in both low-pass and high-pass frames.

At block 804(11), the process evaluates whether this is the last frame. If not, the process continues with the next frame F_{i+1} (block 804(12)).

At block 806, the low-pass frames are sub-sampled at even frames to obtain temporal low-pass frames and the high-pass frames are sub-sampled at odd frames to obtain temporal high-pass frames. If more temporal decomposition levels are desired (i.e., the “yes” branch from block 808), the operations of blocks 802-806 are repeated for the low-pass frames. Note that the motion vectors from frame F_k

1 to F_{k+2} can be obtained by adding the motion vectors from F_k to F_{k+1} and F_{k+1} to
2 F_{k+2} .

3 Following the temporal transform, at block 810, the transformer 130
4 performs spatial 2-D SA-DWT transforms according to the spatial shapes for
5 every temporally transformed frame. This is essentially the same operation
6 illustrated as block 206 in Fig. 2.

7 8 **ESCOT (Block 208)**

9 After wavelet transformation, the resulting wavelet coefficients are coded
10 using a powerful and flexible entropy coding process called ESCOT (Embedded
11 Sub-band Coding with Optimized Truncation) that uses motion information. The
12 entropy coding technique used in ESCOT is similar to the EBCOT (Embedded
13 Block Coding with Optimized Truncation) for still images, which was adopted in
14 JPEG-2000. However, unlike EBCOT, the ESCOT coding scheme is designed for
15 video content and employs a set of coding contexts that make it very suitable for
16 scalable video object compression and the 3D SA-DWT described above, and that
17 take into account motion information between frames. The ESCOT coding
18 scheme is implemented, for example, by the entropy coder 132 of video encoder
19 124 (Fig. 1).

20 The ESCOT coding scheme can be characterized as two main stages: (1)
21 sub-band or entropy coding and (2) bitstream construction. These two stages are
22 described separately below.
23
24
25

Stage 1: Sub-Band Coding

As explained above, the 3-D wavelet transform produces multiple sub-bands of wavelet coefficients. The spatial 2-D wavelet transform decomposes a frame in the horizontal direction and in the vertical direction to produce four sub-bands: a low-low (LL) sub-band, a high-low (HL) sub-band, a low-high (LH) sub-band, and a high-high (HH) sub-band. Fig. 9 shows the four sub-bands from the spatial 2-D wavelet transform. The LL sub-band typically contains the most interesting information. It can be decomposed a second time to produce sub-sub-bands within the LL sub-band, as depicted by bands LL2, LH2, HL2, and HH2.

The ESCOT coding scheme codes each sub-band independently. This is advantageous in that each sub-band can be decoded independently to achieve flexible spatial and temporal scalabilities. A user can mix arbitrary number of spatio-temporal sub-bands in any order to obtain the desired spatial and temporal resolution. Another advantage is that rate-distortion optimization can be done among sub-bands, which may improve compression efficiency.

Fig. 10 shows the sub-band coding process 1000, which is implemented by the entropy coder 132. The process may be implemented as computer-readable instructions that, when executed, perform the operations identified in the sub-band coding process 1000.

At block 1002, the number of contexts used in the coding is reduced by exploiting the symmetric property of wavelet sub-bands through transposition of selected sub-bands. Transposing allows certain sub-bands to share the same context. For example, the LLH sub-band, HLL sub-band, and LHL sub-band that are produced from the 3-D transform can share the same contexts and coding scheme if the HLL and LHL sub-bands are transposed to have the same

1 orientation as the LLH sub-band before encoding. After sub-band transposition,
2 four classes of sub-bands remain: LLL, LLH, LHH and HHH.

3 At block 1004, for each sub-band, the quantized coefficients are coded bit-
4 plane by bit-plane. In a given bit-plane, different coding primitives are used to
5 code a sample's information of this bit-plane. The coding primitives take into
6 account motion information by examining neighboring samples in previous,
7 current, and next frames and determining the significance of these neighboring
8 samples.

9 In one implementation, there are three coding primitives: zero coding (ZC),
10 sign coding (SC) and magnitude refinement (MR). The zero and sign coding
11 primitives are used to code new information for a single sample that is not yet
12 significant in the current bit-plane. Magnitude refinement is used to code new
13 information of a sample that is already significant. Let $\sigma[i,j,k]$ be a binary-valued
14 state variable, which denotes the significance of the sample at position $[i,j,k]$ in the
15 transposed sub-band. The variable $\sigma[i,j,k]$ is initialized to 0 and toggled to 1 when
16 the corresponding sample's first non-zero bit-plane value is coded. Additionally, a
17 variable $\chi[i,j,k]$ is defined as the sign of that sample, which is 0 when the sample
18 is positive and 1 when the sample is negative.

19 **Zero Coding:** When a sample is not yet significant in the previous bit-
20 plane, i.e. $\sigma[i,j,k]=0$, this primitive operation is used to code the new information
21 about the sample. It tells whether the sample becomes significant or not in the
22 current bit-plane. The zero coding operation uses the information of the current
23 sample's neighbors as the context to code the current sample's significance
24 information.

1 More specifically, the zero coding operation evaluates four categories of a
2 sample's neighbors:

- 3
4 1. Immediate horizontal neighbors. The number of horizontal neighbors
5 that are significant are denoted by the variable "h", where $0 < h < 2$.
- 6 2. Immediate vertical neighbors. The number of vertical neighbors that are
7 significant are denoted by the variable "v", where $0 < v < 2$.
- 8 3. Immediate temporal neighbors. The number of temporal neighbors that
9 are significant are denoted by the variable "a", where $0 < a < 2$.
- 10 4. Immediate diagonal neighbors. The number of diagonal neighbors that
11 are significant are denoted by the variable "d", where $0 < d < 12$.

12
13 Fig. 11 shows the four categories of neighbors in three consecutive frames
14 1100, 1102, and 1104. A current sample "s" resides in the middle frame 1102.
15 Two horizontal neighbors "h" reside immediately adjacent to the sample "s" in the
16 middle frame 1102. Two vertical neighbors "v" reside immediately above and
17 below the sample "s" in the middle frame 1102. Two temporal neighbors "a"
18 reside immediately before and after the sample "s" in the previous and following
19 frames 1100 and 1104. Twelve possible diagonal neighbors "d" reside diagonally
20 from the sample "s" in all three frames 1100, 1102, and 1104.

21 It is noted that the temporal neighbors "a" of the sample are not defined as
22 the samples that have the same *spatial* positions in the previous and next frames.
23 Rather, two samples in consecutive frames are deemed to be temporal neighbors
24 when they are in the same motion trajectory. That is, the temporal neighbors are
25 linked by the motion vectors, as illustrated by vectors 1110 and 1112 in Fig. 11.

LLL and LLH Sub-bands				
h	v	a	d	Context
2	x	x	x	0
1	≥ 1	x	x	0
1	0	≥ 1	x	1
1	0	0	x	2
0	2	0	x	3
0	1	0	x	4
0	0	≥ 1	x	5
0	0	0	3	6
0	0	0	2	7
0	0	0	1	8
0	0	0	0	9

LHH Sub-band			
h	v+a	d	Context
2	x	x	0
1	≥ 3	x	0
1	≥ 1	≥ 4	1
1	≥ 1	x	2
1	0	≥ 4	3
1	0	x	4
0	≥ 3	x	5
0	≥ 1	≥ 4	6
0	≥ 1	x	7
0	0	≥ 4	8
0	0	x	9

HHH Sub-band		
d	h+v+a	Context
≥ 6	x	0
≥ 4	≥ 3	1
≥ 4	x	2
≥ 2	≥ 4	3
≥ 2	≥ 2	4
≥ 2	x	5
≥ 0	≥ 4	6
≥ 0	≥ 2	7
≥ 0	1	8
≥ 0	0	9

Tables 1-3: Exemplary Context Assignment map for Zero Coding

Sign Coding: Once a sample becomes significant in the current bit-plane, the sign coding operation is called to code the sign of the significant sample. Sign coding utilizes an adaptive context-based arithmetic coder to compress the sign symbols. Three quantities for the temporal neighbors “a”, the vertical neighbors “v”, and the horizontal neighbors “h” are defined as follows:

$$h = \min\{1, \max\{-1, \sigma[i-1,j,k] \cdot (1-2\chi[i-1,j,k]) + \sigma[i+1,j,k] \cdot (1-2\chi[i+1,j,k])\}\}$$

$$v = \min\{1, \max\{-1, \sigma[i,j-1,k] \cdot (1-2\chi[i,j-1,k]) + \sigma[i,j+1,k] \cdot (1-2\chi[i,j+1,k])\}\}$$

$$a = \min\{1, \max\{-1, \sigma[i,j,k-1] \cdot (1-2\chi[i,j,k-1]) + \sigma[i,j,k+1] \cdot (1-2\chi[i,j,k+1])\}\}$$

The symbol $\hat{\chi}$ means the sign symbol prediction in a given context. The symbol sent to the arithmetic coder is $\hat{\chi}$ XOR χ . An exemplary context assignment map for sign coding of the four sub-bands is provided in Tables 4-6.

h=-1				H=0			
v	a	$\hat{\chi}$	Context	v	a	$\hat{\chi}$	Context
-1	-1	0	0	-1	-1	0	9
-1	0	0	1	-1	0	0	10
-1	1	0	2	-1	1	0	11
0	-1	0	3	0	-1	0	12
0	0	0	4	0	0	0	13
0	1	0	5	0	1	1	12
1	-1	0	6	1	-1	1	11
1	0	0	7	1	0	1	10
1	1	0	8	1	1	1	9

h=1			
v	a	$\hat{\chi}$	Context
-1	-1	1	8
-1	0	1	7
-1	1	1	6
0	-1	1	5
0	0	1	4
0	1	1	3
1	-1	1	2
1	0	1	1
1	1	1	0

Tables 4-6: Exemplary Context Assignment map for Sign Coding

Magnitude Refinement: Magnitude refinement is used to code any new information of a sample that has already become significant in the previous bit-plane. This operation has three possible contexts: 0, 1, or 2. The context is 0 if the magnitude refinement operation is not yet used in the sample. The context is 1 if the magnitude refinement operation has been used in the sample and the sample has at least one significant neighbor. Otherwise, the context is 2.

Using the three coding primitive operations—zero coding, sign coding, and magnitude refinement—a sub-band coefficient can be coded without loss. One preferred implementation of the coding operation 1004 is illustrated in Fig. 10 as blocks 1004(1)-1004(6).

At block 1004(1) in Fig. 10, a significant map is initialized to indicate that all samples are insignificant. As an example, a binary value “1” represents that a sample is significant and a binary value “0” represents that a sample is insignificant. Accordingly, following initialization, the significant map contains all zeros.

Then, for each bit-plane and beginning with the most significant bit-plane, the coding procedure makes three consecutive passes. Each pass processes a “fractional bit-plane”. The reason for introducing multiple coding passes is to ensure that each sub-band has a finely embedded bitstream. By separating zero coding and magnitude refinement into different passes, it is convenient to design efficient and meaningful context assignment. In each pass, the scanning order is along i-direction firstly, then j-direction, and k-direction lastly.

At block 1004(2), a significant propagation pass is performed. This pass processes samples that are not yet significant but have a “preferred neighborhood”,

1 meaning that the sample has at least a significant immediate diagonal neighbor for
2 the HHH sub-band, or at least a significant horizontal, vertical, or temporal
3 neighbor for the other sub-bands. If a sample satisfies these conditions, the zero
4 coding primitive is applied to code the symbol of the current bit-plane for this
5 sample. If the sample becomes significant in the current bit-plane, the sign coding
6 primitive is used to code the sign.

7 At block 1004(3), a magnitude refinement pass is performed to code those
8 samples that are already deemed to be significant. The symbols of these samples
9 in the current bit-plane are coded by the magnitude refinement primitive given
10 above.

11 At block 1004(4), a normalization pass is performed to code those samples
12 that are not yet coded in the previous two passes. These samples are considered
13 insignificant, so zero coding and sign coding primitives are applied in the
14 normalization pass.

15 At block 1004(5), the significant map is updated according to the passes.
16 The updated map reflects the change to those samples that were marked as
17 significant during the passes. Once a sample is identified as significant, it remains
18 significant. This process is then repeated for each bit plane until the least
19 significant bit plane has been coded, as represented by blocks 1004(6) and
20 1004(7).

21 22 Stage 2: Bitstream Construction

23 In the previous stage of sub-band entropy coding, a bitstream is formed for
24 each sub-band. In the 2D realm, there are seven bitstreams; in the 3-D realm,
25 there are fifteen bitstreams. Afterwards, in the current stage, a final bitstream is

constructed by truncating and multiplexing the sub-band bitstreams. The goal is to produce a final bitstream that contains the most effective, yet fewest number, of bits to reduce the amount of data being sent over the network to the receiving client. The bitstream construction takes in consideration that not all decoders will have the same capabilities to decode video. The issue thus becomes how to determine where a bitstream should be truncated and how to multiplex the bitstreams to achieve more functionality (e.g., better PSNR scalability and resolution scalability).

Fig. 12 shows an optimal bitstream truncation and construction procedure 1200, which may be implemented by the entropy coder 132 of the video encoder 124 (Fig. 1). At block 1202, the entropy coder truncates each sub-band bitstream using rate distortion optimization. Given a specific bit-rate R_{\max} , a bitstream can be constructed that satisfies the bit-rate constraint and with minimal distortion. One candidate truncation point is the end of each entropy coding pass. At the end of each pass, the bit length and the distortion reduction is calculated and a value for each candidate truncation point can be plotted to produce an approximate R-D (rate-distortion) curve.

Fig. 13 shows an exemplary R-D curve 1300 formed by five candidate truncation points 1302.

The entropy coder locates the convex hull of the R-D curve, and truncation is performed on those candidate truncation points that reside at the convex hull of R-D curve. This guarantees that at every truncation point, the bitstream is rate-distortion optimized. Given a rate-distortion slope threshold λ , one can find truncation points of a sub-band where the rate-distortion slope is greater than λ . To satisfy the bit-rate constraint and to make the distortion minimal, the smallest

1 value of λ such that $R_\lambda \leq R_{\max}$ is chosen. One suitable algorithm for finding such a
2 threshold can be found in D. Taubman (editor), "JPEG2000 Verification Model:
3 Version VM4.1," ISO/IEC JTC 1/SC 29/WG1 N1286.

4 At block 1204, the entropy coder employs a multi-layer bitstream
5 construction technique to form a final multi-layer bitstream containing a quality
6 level's data. To make a N-layer bitstream, a set of thresholds $\lambda_1 > \lambda_2 > \dots > \lambda_N$ that
7 satisfy $R_{\lambda_N} \leq R_{\max}$ are selected. With every threshold, a truncation point is found
8 and a layer of bitstream from each sub-band is obtained. The corresponding layers
9 from all the sub-bands constitute the layers of the final bitstream.

10 The bitstream construction process offers many advantages in terms of
11 quality scalability, resolution scalability, temporal scalability, and other forms of
12 scalability. The multi-layer bitstream promotes quality scalability in that the
13 client-side decoder 152, depending upon available bandwidth and computation
14 capability, can select one or more layers to be decoded. The fractional bit-plane
15 coding ensures that the bitstream is embedded with fine granularity.

16 Since each sub-band is coded independently, the bitstream of each sub-
17 band is separable. The decoder 152 can easily extract only a few sub-bands and
18 decode only these sub-bands, making resolution scalability and temporal
19 scalability natural and easy. According to the requirement of various multimedia
20 applications, the final bitstream can be constructed in an order to meet the
21 requirement. To obtain resolution or temporal (frame rate) scalability, for
22 example, the bitstream can be assembled sub-band by sub-band, with the lower
23 resolution or lower temporal sub-band in the beginning. For seven sub-bands
24 illustrated in Fig. 9, the four lower level sub-bands can be coded first, followed by
25 the three higher level sub-bands.

Moreover, the final bitstream can be rearranged to achieve other scalability easily because the offset and the length of each layer of bitstream from each sub-band are coded in the header of the bitstream. This property makes the final bitstream very flexible to be re-used for all sorts of applications without re-encoding again.

Conclusion

Although the description above uses language that is specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the invention.

1 **CLAIMS**

2 1. A method comprising:
3 transforming frames in a video sequence using a wavelet transform and
4 motion information between frames to produce multiple sub-bands of coefficients;
5 and
6 coding the coefficients of each sub-band independently.

7
8 2. A method as recited in claim 1, wherein the wavelet transform
9 comprises a shape-adaptive discrete wavelet transform.

10
11 3. A method as recited in claim 1, wherein the transforming comprises
12 performing a temporal 1-D wavelet transform along motion trajectories in a
13 temporal direction.

14
15 4. A method as recited in claim 1, wherein the transforming comprises:
16 performing a temporal wavelet transform on corresponding pixels in a
17 video object along motion trajectories in a temporal direction to produce frames of
18 temporal wavelet coefficients; and
19 performing a spatial wavelet transform on the frames of the temporal
20 wavelet coefficients to produce multiple sub-bands of wavelet coefficients.

21
22 5. A method as recited in claim 1, wherein the coding produces multiple
23 bitstreams, one for each sub-band, and further comprising forming a bitstream
24 from the multiple bitstreams.
25

1 6. A method as recited in claim 1, wherein the coding comprises
2 transposing selected sub-bands.

3
4 7. A method as recited in claim 1, wherein the coding comprises coding
5 the coefficients of each sub-band bit-plane by bit-plane using different coding
6 primitives.

7
8 8. A method as recited in claim 7, wherein the coding primitives
9 comprise:

10 zero coding to code new information about a coefficient that is not yet
11 significant in a previous bit-plane; and

12 sign coding to code a sign of the coefficient once the coefficient is deemed
13 significant.

14
15 9. A method as recited in claim 7, wherein the coding primitives
16 comprise:

17 zero coding to code new information about a coefficient that is not yet
18 significant in a previous bit-plane;

19 sign coding to code a sign of the coefficient once the coefficient is deemed
20 significant; and

21 magnitude refinement to code new information of a coefficient that has
22 already become significant in the previous bit-plane.

1 10. A method as recited in claim 1, wherein the coding comprises
2 assigning contexts to the coefficients of each sub-band based on numbers of
3 significant neighboring samples.

4
5 11. A method as recited in claim 10, wherein the sub-bands include an
6 LLL (low-low-low) sub-band and an LLH (low-low-high) sub-band and the
7 contexts are assigned as follows:

8

LLL and LLH Sub-bands				
h	v	a	d	Context
2	x	x	x	0
1	≥ 1	x	x	0
1	0	≥ 1	x	1
1	0	0	x	2
0	2	0	x	3
0	1	0	x	4
0	0	≥ 1	x	5
0	0	0	3	6
0	0	0	2	7
0	0	0	1	8
0	0	0	0	9

16

17
18 where "h" represents a number of immediate horizontal neighbors that are
19 significant and $0 < h < 2$, "v" represents a number of immediate vertical neighbors
20 that are significant and $0 < v < 2$, "a" represents a number of immediate temporal
21 neighbors that are significant and $0 < a < 2$, and "d" represents a number of
22 immediate diagonal neighbors that are significant and $0 < d < 12$.

12. A method as recited in claim 10, wherein the sub-bands include an LHH (low-high-high) sub-band and the contexts are assigned as follows:

LHH Sub-band			
h	v+a	d	Context
2	x	x	0
1	≥ 3	x	0
1	≥ 1	≥ 4	1
1	≥ 1	x	2
1	0	≥ 4	3
1	0	x	4
0	≥ 3	x	5
0	≥ 1	≥ 4	6
0	≥ 1	x	7
0	0	≥ 4	8
0	0	x	9

where “h” represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, “a” represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and “d” represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

1 **13.** A method as recited in claim 10, wherein the sub-bands include an
2 HHH (high-high-high) sub-band and the contexts are assigned as follows:

3

d	h+v+a	Context
≥ 6	x	0
≥ 4	≥ 3	1
≥ 4	x	2
≥ 2	≥ 4	3
≥ 2	≥ 2	4
≥ 2	x	5
≥ 0	≥ 4	6
≥ 0	≥ 2	7
≥ 0	1	8
≥ 0	0	9

10

11

12 where “h” represents a number of immediate horizontal neighbors that are
13 significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors
14 that are significant and $0 < v < 2$, “a” represents a number of immediate temporal
15 neighbors that are significant and $0 < a < 2$, and “d” represents a number of
16 immediate diagonal neighbors that are significant and $0 < d < 12$.

17

18 **14.** A method as recited in claim 1, further comprising truncating a
19 number of bits in each bit-plane according to rate-distortion curves.

20

21 **15.** A method as recited in claim 1, further comprising estimating
22 motion trajectories of pixels in a video object from frame to frame in the video
23 sequence and said transforming is performed on corresponding pixels along the
24 motion trajectories.

25

1 **16.** A computer-readable medium comprising computer-executable
2 instructions that, when executed by one or more processors, perform the method as
3 recited in claim 1.

4
5 **17.** A method comprising:
6 estimating motion trajectories of pixels in a video object from frame to
7 frame in a video sequence;
8 performing a temporal wavelet transform on the corresponding pixels along
9 the motion trajectories in a temporal direction to produce frames of temporal
10 wavelet coefficients;
11 performing a spatial wavelet transform on the frames of the temporal
12 wavelet coefficients to produce multiple sub-bands of wavelet coefficients; and
13 coding each sub-band of wavelet coefficients independently.

14
15 **18.** A method as recited in claim 17, wherein the estimating comprises
16 matching corresponding pixels in the video object from frame to frame in the
17 video sequence.

18
19 **19.** A method as recited in claim 17, wherein the temporal and spatial
20 wavelet transforms comprise a shape-adaptive discrete wavelet transform.

21
22 **20.** A method as recited in claim 17, wherein the performing a temporal
23 wavelet transform comprises:

24 forming a pixel array containing pixels that continue from frame to frame in
25 the video sequence;

1 examining a pixel in a frame to determine whether the pixel is a terminating
2 pixel that does not continue to a next frame;

3 if the pixel is a terminating pixel, terminating the pixel array; and

4 if the pixel is not a terminating pixel, adding the pixel to the pixel array.

5
6 **21.** A method as recited in claim 20, further comprising transforming
7 the pixels arrays to produce the frames of temporal wavelet coefficients.

8
9 **22.** A method as recited in claim 17, wherein the coding comprises
10 transposing selected sub-bands to reduce a number of sub-bands to be coded.

11
12 **23.** A method as recited in claim 17, wherein the coding comprises:
13 coding the wavelet coefficients in bit-planes; and
14 allocating bits for the bit-planes according to a rate-distortion optimization.

15
16 **24.** A method as recited in claim 17, further comprising truncating bits
17 allocated to a bit-plane at a point on a rate-distortion curve that approximates a
18 convex hull.

19
20 **25.** A method as recited in claim 17, wherein the coding comprises
21 coding the wavelet coefficients of each sub-band bit-plane by bit-plane using
22 different coding primitives.

001290"09T5550

1 **30.** A computer-readable medium comprising computer-executable
2 instructions that, when executed by one or more processors, perform the method as
3 recited in claim 17.

4
5 **31.** A method comprising:
6 forming an array containing a current pixel in a current frame;
7 determining whether a pixel corresponding to the current pixel resides in a
8 next frame subsequent to the current frame;
9 if the corresponding pixel exists, add the current pixel to the array; and
10 if the corresponding pixel does not exist, terminate the pixel array with the
11 current pixel.

12
13 **32.** A method as recited in claim 31, further comprising initially
14 indicating all pixels as one state and changing the state of each pixel after said
15 examining of the pixel.

16
17 **33.** A method as recited in claim 31, further comprising applying a
18 wavelet transform to the pixel array to produce wavelet coefficients.

19
20 **34.** A method as recited in claim 31, further comprising:
21 applying a wavelet transform to the pixel array to produce wavelet
22 coefficients; and
23 applying a spatial 2-D wavelet transform on the wavelet coefficients.
24
25

1 **35.** A computer-readable medium comprising computer-executable
2 instructions that, when executed by one or more processors, perform the method as
3 recited in claim 31.

4
5 **36.** A method comprising:
6 coding sub-bands of coefficients produced from transforming video frames
7 in an independent manner such that one sub-band of coefficients is coded
8 independently of another sub-band of coefficients; and
9 constructing a bitstream from the independently coded sub-bands.

10
11 **37.** A method as recited in claim 36, wherein the coding comprises
12 transposing selected sub-bands prior to said coding.

13
14 **38.** A method as recited in claim 36, wherein the coding comprises
15 coding the coefficients of each sub-band bit-plane by bit-plane using different
16 coding primitives.

17
18 **39.** A method as recited in claim 38, wherein the coding primitives
19 comprise:

20 zero coding to code new information about a coefficient that is not yet
21 significant in a previous bit-plane; and

22 sign coding to code a sign of the coefficient once the coefficient is deemed
23 significant.

1 **40.** A method as recited in claim 38, wherein the coding primitives
2 comprise:

3 zero coding to code new information about a coefficient that is not yet
4 significant in a previous bit-plane;

5 sign coding to code a sign of the coefficient once the coefficient is deemed
6 significant; and

7 magnitude refinement to code new information of a coefficient that has
8 already become significant in the previous bit-plane.

9
10 **41.** A method as recited in claim 36, wherein the coding comprises
11 assigning contexts to the coefficients of each sub-band based on numbers of
12 significant neighboring samples.

13
14 **42.** A method as recited in claim 41, wherein the sub-bands include an
15 LLL (low-low-low) sub-band and an LLH (low-low-high) sub-band and the
16 contexts are assigned as follows:

LLL and LLH Sub-bands				
h	v	a	d	Context
2	x	x	x	0
1	≥ 1	x	x	0
1	0	≥ 1	x	1
1	0	0	x	2
0	2	0	x	3
0	1	0	x	4
0	0	≥ 1	x	5
0	0	0	3	6
0	0	0	2	7
0	0	0	1	8
0	0	0	0	9

where “h” represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, “a” represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and “d” represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

43. A method as recited in claim 41, wherein the sub-bands include an LHH (low-high-high) sub-band and the contexts are assigned as follows:

LHH Sub-band			
h	v+a	d	Context
2	x	x	0
1	≥ 3	x	0
1	≥ 1	≥ 4	1
1	≥ 1	x	2
1	0	≥ 4	3
1	0	x	4
0	≥ 3	x	5
0	≥ 1	≥ 4	6
0	≥ 1	x	7
0	0	≥ 4	8
0	0	x	9

where “h” represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, “a” represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and “d” represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

1 **44.** A method as recited in claim 41, wherein the sub-bands include an
2 HHH (high-high-high) sub-band and the contexts are assigned as follows:

3

d	h+v+a	Context
≥ 6	x	0
≥ 4	≥ 3	1
≥ 4	x	2
≥ 2	≥ 4	3
≥ 2	≥ 2	4
≥ 2	x	5
≥ 0	≥ 4	6
≥ 0	≥ 2	7
≥ 0	1	8
≥ 0	0	9

10

11
12 where "h" represents a number of immediate horizontal neighbors that are
13 significant and $0 < h < 2$, "v" represents a number of immediate vertical neighbors
14 that are significant and $0 < v < 2$, "a" represents a number of immediate temporal
15 neighbors that are significant and $0 < a < 2$, and "d" represents a number of
16 immediate diagonal neighbors that are significant and $0 < d < 12$.

17
18 **45.** A method as recited in claim 36, wherein the constructing comprises
19 forming multiple bit-planes and truncating a number of bits in each bit-plane
20 according to a rate-distortion curve.

21
22 **46.** A computer-readable medium comprising computer-executable
23 instructions that, when executed by one or more processors, perform the method as
24 recited in claim 36.
25

007390-09T65660

1 **47.** A method comprising:
2 transforming a video sequence to produce a set of sub-bands;
3 transposing selected sub-bands to produce a reduced set of sub-bands that
4 are fewer than the set of sub-bands; and
5 coding the reduced set of sub-bands.

6
7 **48.** A method as recited in claim 47, wherein the coding comprises
8 coding each sub-band independently.

9
10 **49.** A method as recited in claim 47, wherein the coding comprises
11 assigning contexts to the coefficients of each sub-band based on numbers of
12 significant neighboring samples.

13
14 **50.** A computer-readable medium comprising computer-executable
15 instructions that, when executed by one or more processors, perform the method as
16 recited in claim 47.

17
18 **51.** A method for coding coefficients indicative of transformed video in
19 multiple bit-planes, comprising:
20 conducting coding passes through a bit-plane to code significant samples
21 separately from insignificant samples; and
22 repeating said conducting for each bit-plane.

23
24
25

007330"09T6550

1 **57.** A video encoder as recited in claim 54, wherein the wavelet
2 transformer comprises a 3-D wavelet transformer that applies:

3 (1) a temporal 1-D wavelet transform on corresponding pixels in
4 consecutive frames along motion trajectories in a temporal direction to produce
5 temporal wavelet coefficients; and

6 (2) a spatial 2-D wavelet transform on the temporal wavelet coefficients.
7

8 **58.** A video encoder as recited in claim 54, wherein the wavelet
9 transformer estimates motion trajectories of pixels in a video object from frame to
10 frame in the video sequence and initially transforms corresponding pixels along
11 the motion trajectories in the temporal direction.
12

13 **59.** A video encoder as recited in claim 54, wherein the coder codes
14 transposes selected sub-bands to produced reduced set of sub-bands.
15

16 **60.** A video encoder as recited in claim 54, wherein the coder codes the
17 coefficients of each sub-band into bit-planes using different coding primitives.
18

19 **61.** A video encoder as recited in claim 54, wherein the coder comprises
20 a context-based arithmetic coder to assign contexts to the coefficients of each sub-
21 band based on different coding primitives.
22
23
24
25

62. A video encoder as recited in claim 61, wherein the sub-bands include an LLL (low-low-low) sub-band and an LLH (low-low-high) sub-band and the coder employs a zero coding primitive to code new information about a coefficient that is not yet significant in a previous bit-plane by assigning the contexts as follows:

LLL and LLH Sub-bands				
h	v	a	d	Context
2	x	x	x	0
1	≥ 1	x	x	0
1	0	≥ 1	x	1
1	0	0	x	2
0	2	0	x	3
0	1	0	x	4
0	0	≥ 1	x	5
0	0	0	3	6
0	0	0	2	7
0	0	0	1	8
0	0	0	0	9

where “h” represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, “a” represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and “d” represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

63. A video encoder as recited in claim 61, wherein the sub-bands include an LHH (low-high-high) sub-band and the coder employs a zero coding primitive to code new information about a coefficient that is not yet significant in a previous bit-plane by assigning the contexts as follows:

LHH Sub-band			
h	v+a	d	Context
2	x	x	0
1	≥ 3	x	0
1	≥ 1	≥ 4	1
1	≥ 1	x	2
1	0	≥ 4	3
1	0	x	4
0	≥ 3	x	5
0	≥ 1	≥ 4	6
0	≥ 1	x	7
0	0	≥ 4	8
0	0	x	9

where "h" represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, "v" represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, "a" represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and "d" represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

64. A video encoder as recited in claim 61, wherein the sub-bands include an HHH (high-high-high) sub-band and the coder employs a zero coding primitive to code new information about a coefficient that is not yet significant in a previous bit-plane by assigning the contexts as follows:

d	h+v+a	Context
≥ 6	x	0
≥ 4	≥ 3	1
≥ 4	x	2
≥ 2	≥ 4	3
≥ 2	≥ 2	4
≥ 2	x	5
≥ 0	≥ 4	6
≥ 0	≥ 2	7
≥ 0	1	8
≥ 0	0	9

where "h" represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, "v" represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, "a" represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and "d" represents a number of immediate diagonal neighbors that are significant and $0 < d < 12$.

65. A video encoder as recited in claim 61, wherein the coder employs a sign coding primitive to code a sign of the coefficient once the coefficient is deemed significant by assigning the contexts as follows:

h=-1			
v	a	$\hat{\chi}$	Context
-1	-1	0	0
-1	0	0	1
-1	1	0	2
0	-1	0	3
0	0	0	4
0	1	0	5
1	-1	0	6
1	0	0	7
1	1	0	8

H=0			
v	a	$\hat{\chi}$	Context
-1	-1	0	9
-1	0	0	10
-1	1	0	11
0	-1	0	12
0	0	0	13
0	1	1	12
1	-1	1	11
1	0	1	10
1	1	1	9

h=1			
v	a	$\hat{\chi}$	Context
-1	-1	1	8
-1	0	1	7
-1	1	1	6
0	-1	1	5
0	0	1	4
0	1	1	3
1	-1	1	2
1	0	1	1
1	1	1	0

where “h” represents a number of immediate horizontal neighbors that are significant and $0 < h < 2$, “v” represents a number of immediate vertical neighbors that are significant and $0 < v < 2$, “a” represents a number of immediate temporal neighbors that are significant and $0 < a < 2$, and \hat{x} is a sign symbol prediction in a given context.

1 **66.** A video encoder as recited in claim 54, wherein the coder truncates
2 a number of coding bits according to rate-distortion curves.

3
4 **67.** An operating system embodied on a computer-readable medium
5 comprising a video encoder as recited in claim 54.

6
7 **68.** A video encoder comprising:
8 means for estimating motion trajectories of pixels in a video object from
9 frame to frame in a video sequence;
10 means for performing a temporal wavelet transform on the corresponding
11 pixels along the motion trajectories in a temporal direction to produce frames of
12 temporal wavelet coefficients;
13 means for performing a spatial wavelet transform on the frames of the
14 temporal wavelet coefficients to produce multiple sub-bands of wavelet
15 coefficients; and
16 means for coding each sub-band of wavelet coefficients independently.

17
18 **69.** A video encoder as recited in claim 68, wherein the estimating
19 means comprises means for matching corresponding pixels in the video object
20 from frame to frame in the video sequence.

21
22 **70.** A video encoder as recited in claim 68, wherein the temporal and
23 spatial wavelet transforms comprise a shape-adaptive discrete wavelet transform.
24
25

1 71. A video encoder as recited in claim 68, wherein the means for
2 performing a temporal wavelet transform comprises:

3 means for forming a pixel array containing pixels that continue from frame
4 to frame in the video sequence;

5 means for examining a pixel in a frame to determine whether the pixel is a
6 terminating pixel that does not continue to a next frame;

7 if the pixel is a terminating pixel, means for terminating the pixel array; and

8 if the pixel is not a terminating pixel, means for adding the pixel to the
9 pixel array.

10
11 72. A video encoder as recited in claim 68, wherein the coding means
12 comprises means for transposing selected sub-bands to reduce a number of sub-
13 bands to be coded.

14
15 73. A video encoder as recited in claim 68, wherein the coding means
16 comprises:

17 means for coding the wavelet coefficients in bit-planes; and

18 means for allocating bits for the bit-planes according to a rate-distortion
19 optimization.

20
21 74. A video encoder as recited in claim 68, further comprising means
22 for truncating bits allocated to a bit-plane at a point on a rate-distortion curve that
23 approximates a convex hull.

1 75. A video encoder as recited in claim 68, wherein the coding means
2 comprises means for coding the wavelet coefficients of each sub-band bit-plane by
3 bit-plane using different coding primitives.
4

5 76. A video encoder as recited in claim 68, wherein the coding means
6 produces multiple bitstreams for corresponding sub-bands of wavelet coefficients
7 and further comprising means for constructing a multi-layer bitstream from the
8 multiple bitstreams.
9

10 77. A video encoder as recited in claim 68, wherein the coding means
11 comprises means for assigning contexts to the wavelet coefficients of each sub-
12 band based on numbers of significant neighboring samples.
13

14 78. A computer-readable medium comprising computer-executable
15 instructions that, when executed on a processor, direct a device to:

16 code sub-bands of coefficients produced from transforming video frames in
17 an independent manner such that one sub-band of coefficients is coded
18 independently of another sub-band of coefficients; and

19 construct a bitstream from the independently coded sub-bands.
20

21 79. A computer-readable medium as recited in claim 78, further
22 comprising computer-executable instructions that, when executed on a processor,
23 direct a device to transpose selected sub-bands prior to said coding.
24
25

1 **80.** A computer-readable medium as recited in claim 78, further
2 comprising computer-executable instructions that, when executed on a processor,
3 direct a device to code the coefficients of each sub-band bit-plane by bit-plane
4 using different coding primitives.

5
6 **81.** A computer-readable medium as recited in claim 78, further
7 comprising computer-executable instructions that, when executed on a processor,
8 direct a device to assign contexts to the coefficients of each sub-band based on
9 numbers of significant neighboring samples.

10
11 **82.** A computer-readable medium embodying an encoded video signal
12 constructed as a result of a process comprising:

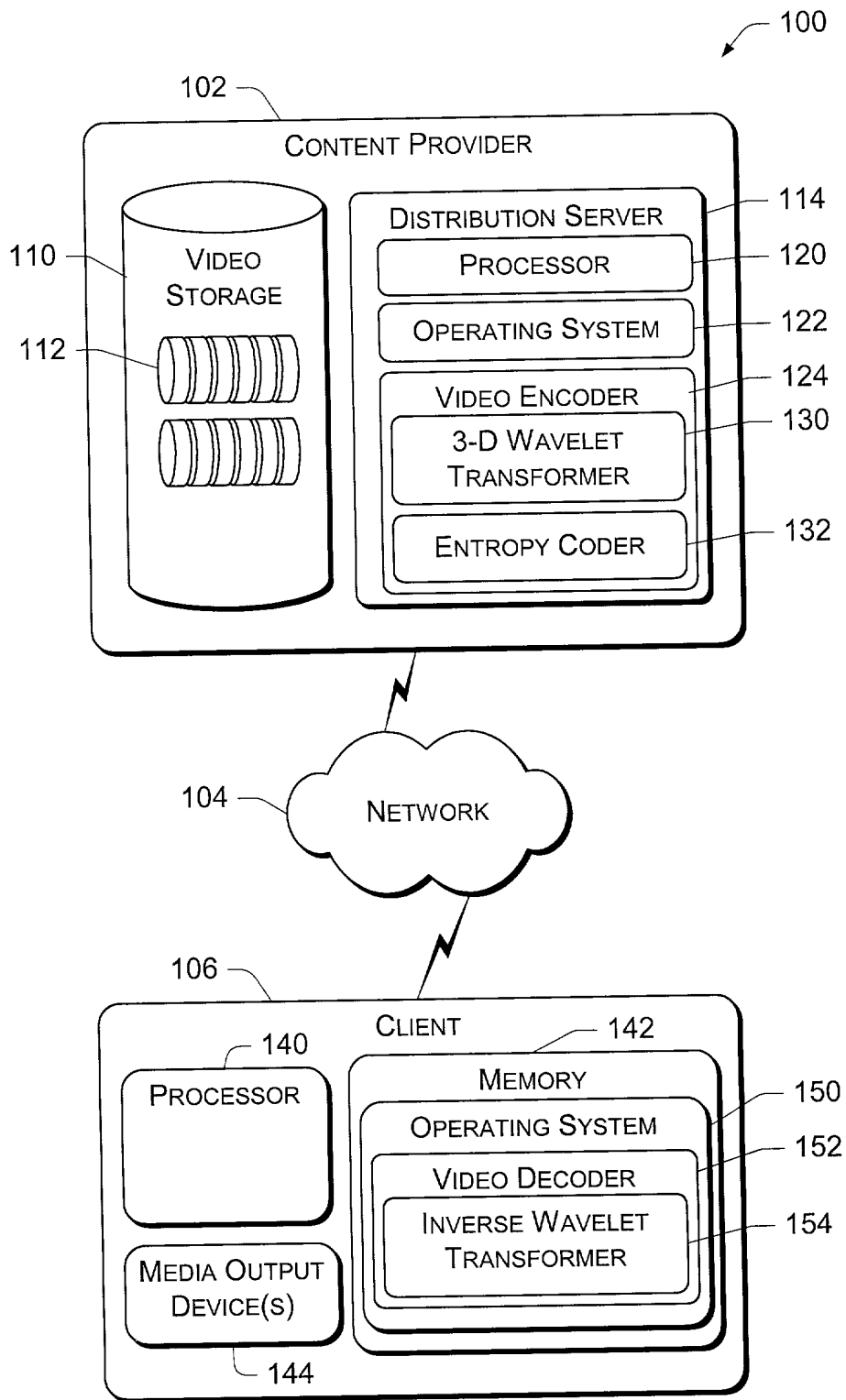
13 transforming frames in a video sequence using a wavelet transform to
14 produce multiple sub-bands of coefficients;

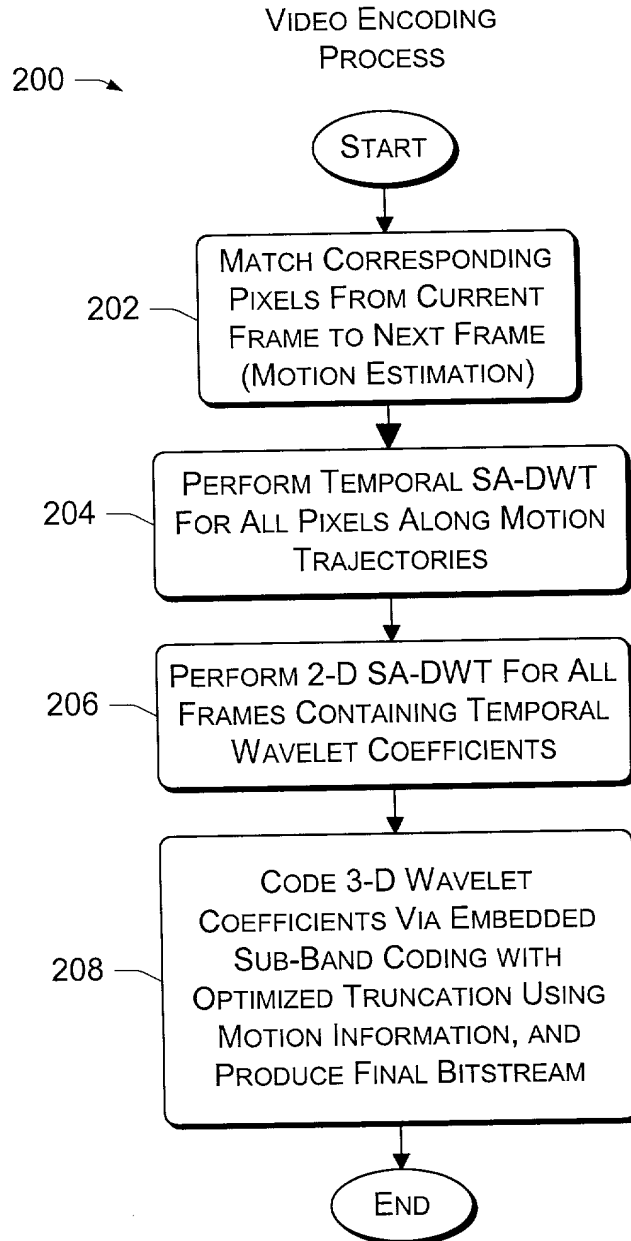
15 coding the coefficients of each sub-band independently to produce multiple
16 bitstreams; and

17 forming a bitstream from the multiple bitstreams.
18
19
20
21
22
23
24
25

1 **ABSTRACT**

2 A video encoding system and method utilizes a three-dimensional (3-D)
3 wavelet transform and entropy coding that utilize motion information in a way to
4 reduce the sensitivity to motion. In one implementation, the coding process
5 initially estimates motion trajectories of pixels in a video object from frame to
6 frame in a video sequence to account for motion of the video object throughout the
7 frames. After motion estimation, a 3-D wavelet transform is applied in two parts.
8 First, a temporal 1-D wavelet transform is applied to the corresponding pixels
9 along the motion trajectories in a time direction. The temporal wavelet transform
10 produces decomposed frames of temporal wavelet transforms, where the spatial
11 correlation within each frame is well preserved. Second, a spatial 2-D wavelet
12 transform is applied to all frames containing the temporal wavelet coefficients.
13 The wavelet transforms produce coefficients within different sub-bands. The
14 process then codes wavelet coefficients. In particular, the coefficients are assigned
15 various contexts based on the significance of neighboring samples in previous,
16 current, and next frame, thereby taking advantage of any motion information
17 between frames. The wavelet coefficients are coded independently for each sub-
18 band to permit easy separation at a decoder, making resolution scalability and
19 temporal scalability natural and easy. During the coding, bits are allocated among
20 sub-bands according to a technique that optimizes rate-distortion characteristics.
21
22
23
24
25

*Fig. 1*

*Fig. 2*

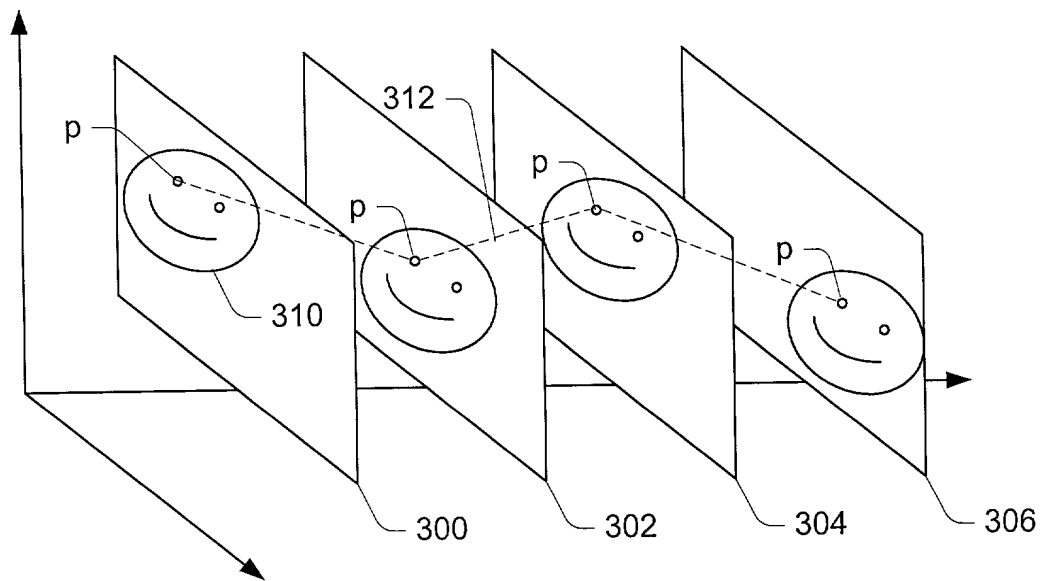


Fig. 3

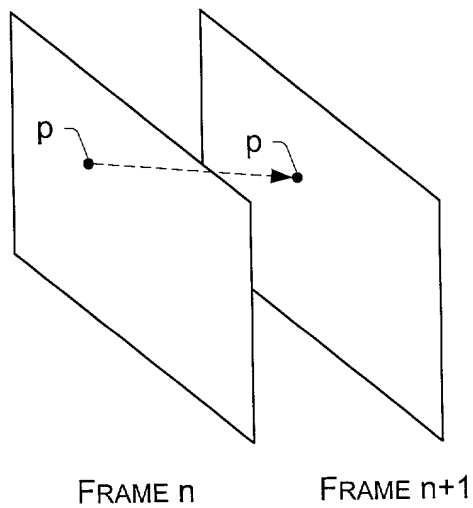


Fig. 4

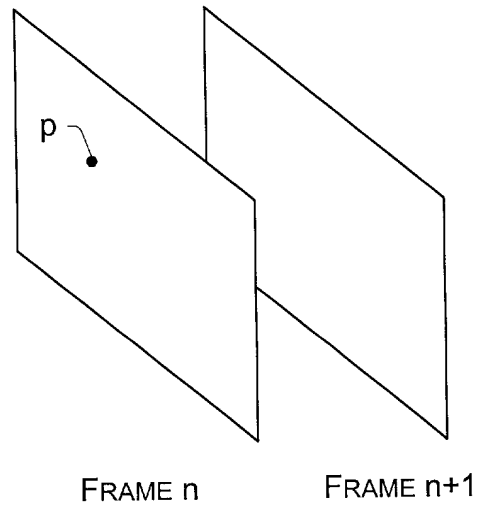


Fig. 5

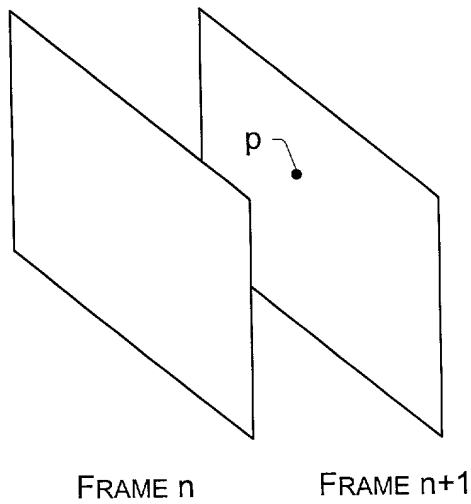


Fig. 6

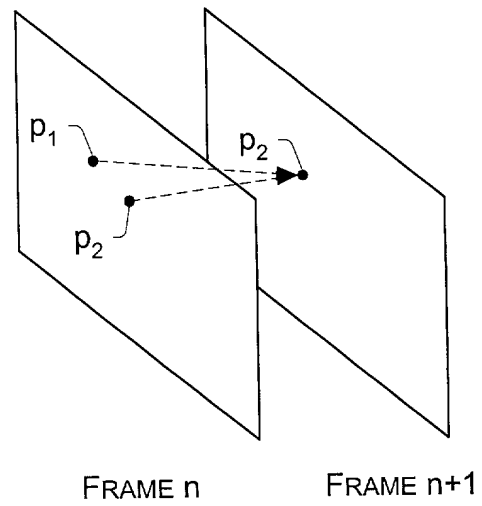
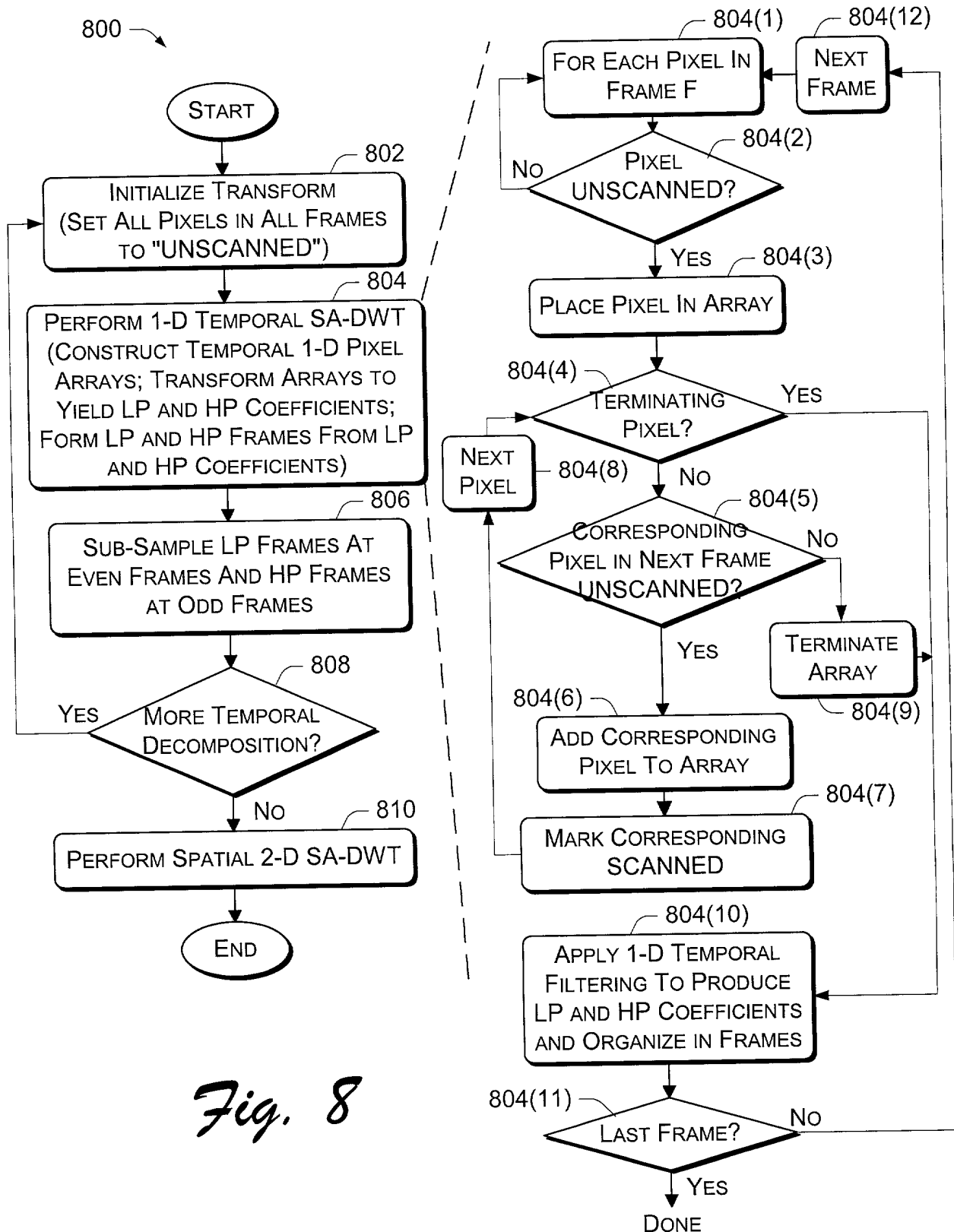


Fig. 7

0959160-06100
001390-0915550

3-D WAVELET TRANSFORM



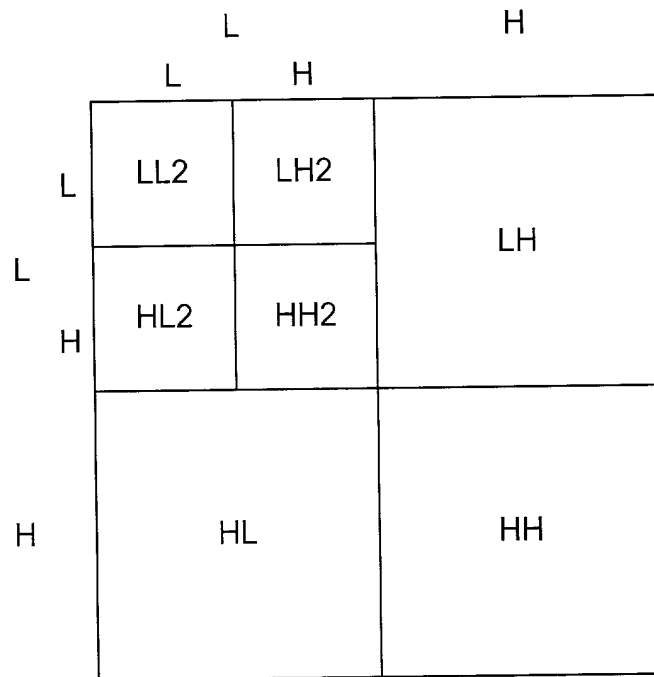


Fig. 9

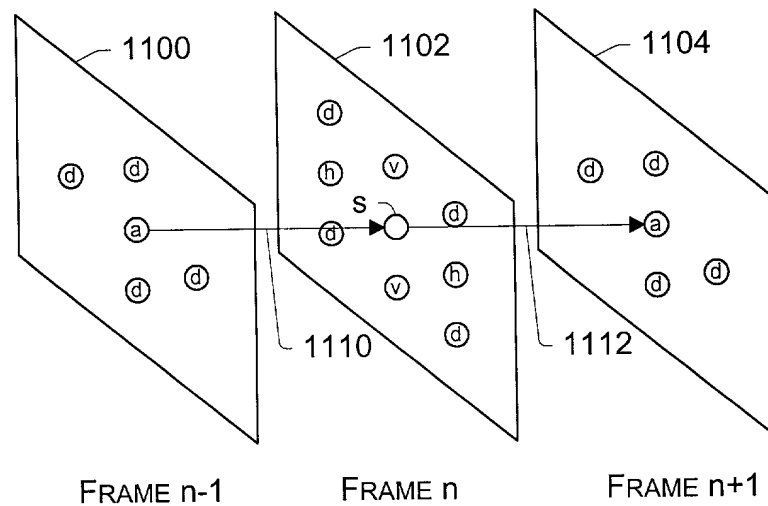


Fig. 11

1000

ESCOT: SUB-BAND ENCODING

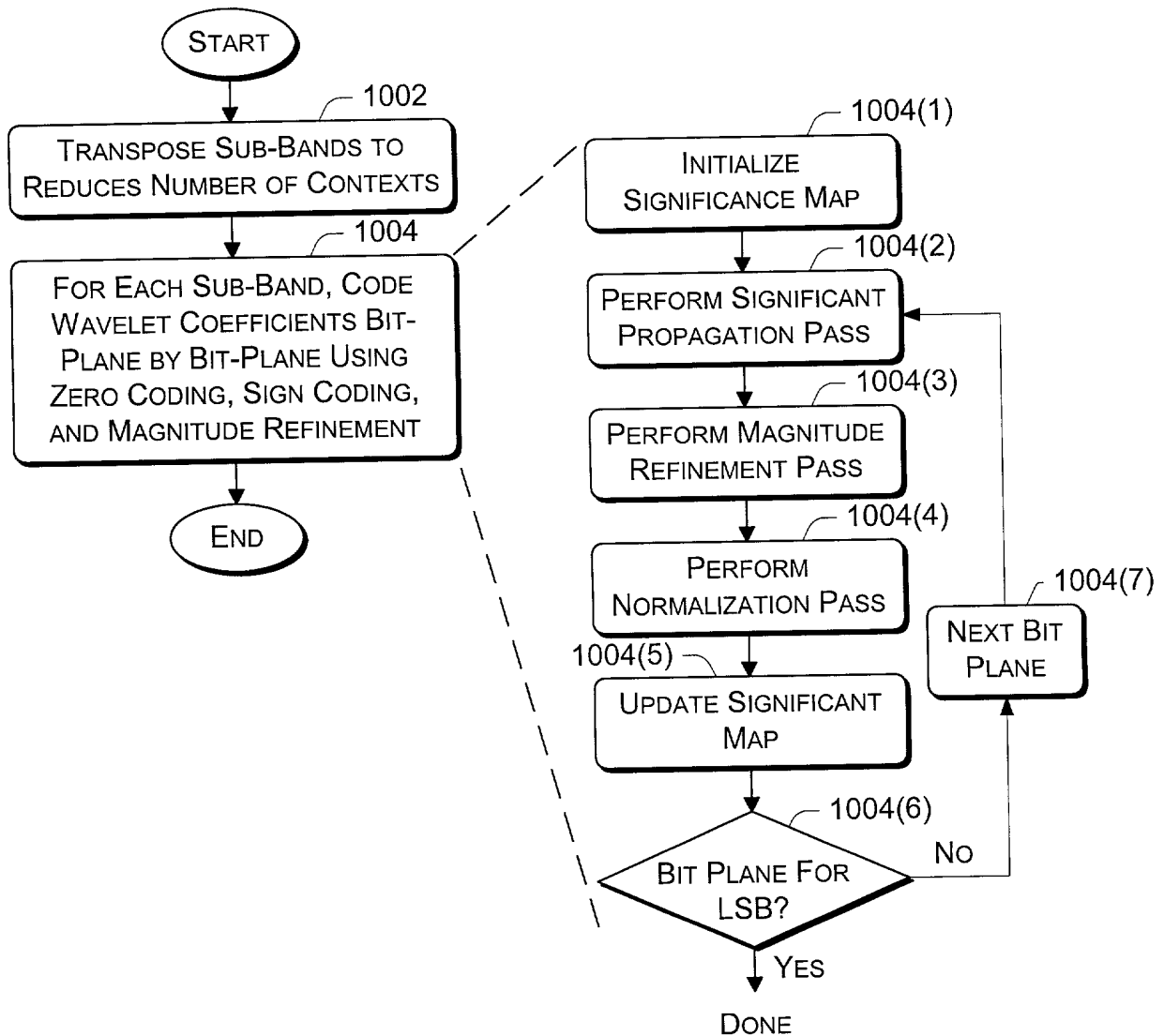


Fig. 10

ESCOT: BITSTREAM
CONSTRUCTION

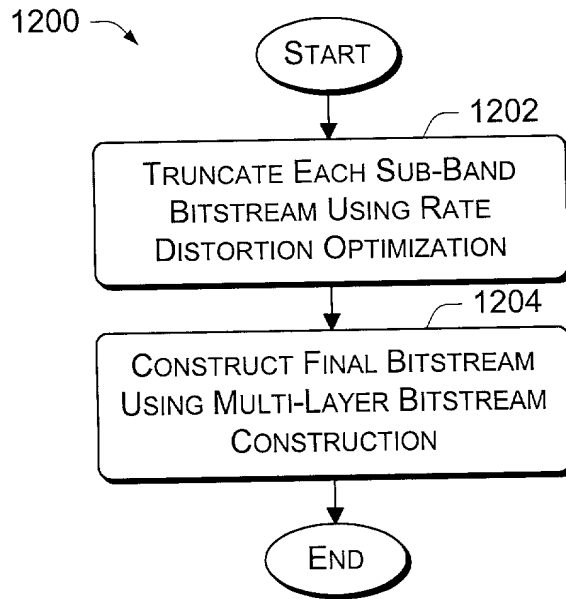


Fig. 12

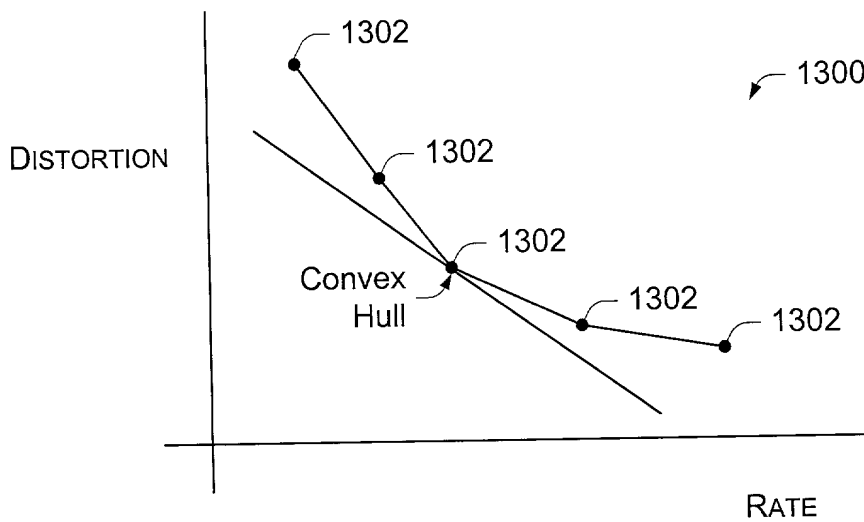


Fig. 13